6-29-2021

Helipad detection from satellite imagery using convolutional neural networks

David Specht
Rowan University

HELIPAD DETECTION FROM SATELLITE IMAGERY USING CONVOLUTIONAL NEURAL NETWORKS

by

David Specht

A Thesis

Submitted to the
Department of Electrical & Computer Engineering
College of Engineering
In partial fulfillment of the requirement
For the degree of
Master of Science in Electrical and Computer Engineering
at
Rowan University
June 15, 2021

Thesis Chair: Nidhal C. Bouaynaya, Ph.D.

Committee Members:
Ghulam Rasool, Ph.D.
Charles C. Johnson, FAA

Acknowledgment

This work was supported by the Federal Aviation Administration (FAA) Cooperative Agreement Number 16- G-015 and NSF Award DUE-1610911. This Thesis was also supported by a subaward from Rutgers University, Center for Advanced Infrastructure & Transportation, under Grant no. 69A3551847102 from the U.S. Department of Transportation, Office of the Assistant Secretary for Research and Technology (OST-R). We would also like to thank LZControl for their guidance and assistance with this effort. Via a Cooperative Research and Development Agreement with the FAA, LZControl provided a set of data from their system and subject matter expertise, which provides landing zones for helicopters across the U.S., often complementing the FAA's 5010 database and including locations/sites not present in the FAA's 5010 system.

Abstract

David Specht
HELIPAD DETECTION FROM SATELLITE IMAGERY USING CONVOLUTIONAL NEURAL NETWORK
2020-2021
Nidhal C. Bouaynaya
Master of Science in Electrical and Computer Engineering

Location data about U.S. heliports is often inaccurate or nonexistent in the FAA's databases, which leaves pilots and air ambulance operators with inaccurate information about where to find safe landing zones. In the 2018 FAA Reauthorization Act, Congress required the FAA to collect better information from the helicopter industry under part 157, which covers the construction, alteration, activation and deactivation of airports and heliports. At the same time, there is no requirement to report private helipads to the FAA when constructed or removed, and some public heliports do not have up to date records. This thesis proposes an autonomous system that can authenticate the coordinates in the FAA master database, as well as search for helipads in a designated large area. The proposed system is based on a convolutional neural network model that learns optimal helipad features from the data. We used the FAA's 5010 database and others to construct a benchmark database of rotorcraft landing sites. The database consists of 9,324 aerial images, containing helipads, helistops, helidecks, and helicopter runways in rural and urban areas, as well as negative examples, such as rooftop buildings and fields. The dataset was then used to train various convolutional neural network models. The outperforming model, EfficientNet-b0, achieved nearly 95% accuracy on the validation set. We subsequently implemented a gradient-based explainability map depicting the most salient pixels that influenced the network's prediction.

Table of Contents

Table of Contents (Continued)

List of Figures

List of Tables

Table Page

Chapter 1

Background and Introduction

I-A Problem Formulation and Motivation

The FAA maintains a database reported helipads. So that pilots can identify possible landing locations, the dataset also contains other information besides the latitude/longitude coordinates for identification, including, in some cases, the information needed to request permission to land at these sites. Using the FAA Forms 7480&5010, lat/lon location coordinates are recorded by the helipad owners and using the same form alterations due to the removal or relocation of the helipad are also reported to the FAA. This helps to keep the dataset accurate as helipads close or relocate and as new ones become operational. However, reporting of these helipads for private use facilities is not required, primarily as they are updated for changed/modified information. Additionally, no ongoing audit process to ensure the accuracy of these coordinates currently exists. Consequently, there are numerous helipads across the U.S. that are not found in the database or that have not been updated over time as they have closed, and various entities have built new helipads in their place. As a result, the accuracy of the database can often be dubious a best as it is known to contain numerous errors and does not contain all possible helipads. This could cause an issue if a pilot flies to a set of coordinates that does not actually contain a helipad, which could result in a fuel exhaustion or other causal factors leading to and accident/incident from a wrong helipad landing.

This thesis proposes a solution to the problem of helipad identification and accuracy validation via the form of an autonomous helipad identification system. With

the large availability of satellite imagery, it becomes possible to collect overhead imagery of desired coordinates. As helipads should be visible from this imagery, this problem now becomes a problem of identifying helipads in imagery. While not entirely accurate, many accurate coordinates can still be used to create a dataset of helipad imagery. This data can then be used to train a system to determine the existence of helipads at specified coordinates and expanded to allow users to search for a helipad within a specified region.

Simply by being able to verify the existence of a helipad at a given coordinate, it becomes possible to identify errors in other datasets, which can be noted to reduce the odds of pilots receiving incorrect information. This system can then be extended to be able to search for locations that contain helipads. This searching ability will allow for new coordinates to be proposed for the database. Depending on the implementation of the search algorithm, it may also be possible to correct the coordinates to improve their accuracy.

A-1 What are Helipads?

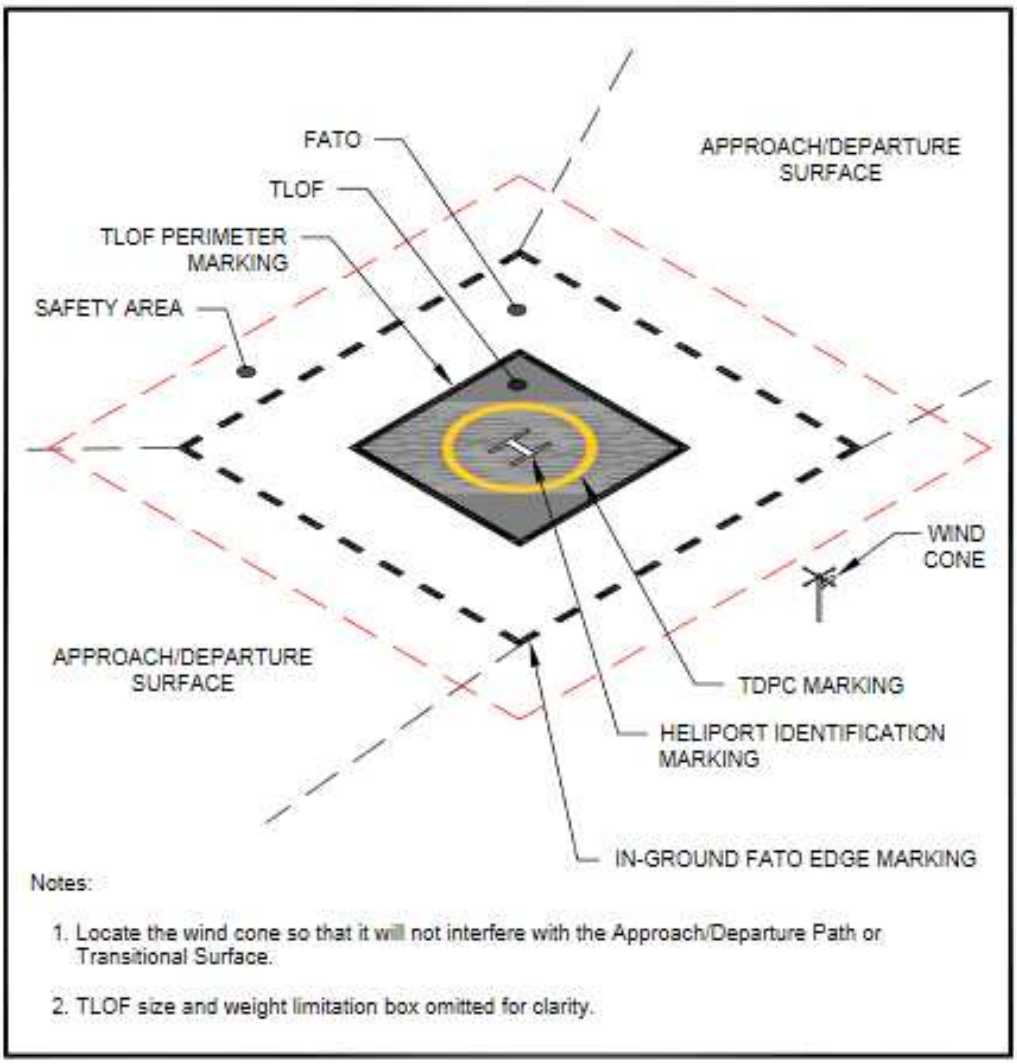
A helipad is a designated area that a helicopter is intended to land on and consisting of the Touch Down and LiftOff area (TLOF) of the landing area. The area around the TLOF, the Final Approach and TakeOff area (FATO) will be a solid surface clear of obstacles and may also be marked as part of the landing surface if it meets certain criteria defined in the FAA's Advisory circular 150/5390-2C. The FAA's Advisory Circular 150/5390-2C [1] defines standards for the construction of helipads.

However, section 203 notes that these standards are not the law and act as recommendations for Prior Permission Required (PPR) facilities. Section 103 effectively

states that the minimum required facilities are a clear area with a wind cone. Despite this, many heliports follow these standards and use a variant of the markings listed in section 215, which includes a white “H” marking in the middle the marked borders of the landing surface. Of note is that as per section 414, all hospital helipads will also include a cross around the “H”, and section 215 allows for PPR facilities to replace the “H” with another distinctive marking such as a logo. The TLOF is the area where a helipad is intended to land, and this area must always be load bearing. The FATO is the area around the TLOF where the helicopter should have room to maneuver. Section 208 notes that this surface does not always need to be load bearing, but the TLOF perimeter needs to be clearly marked in such a case. If the FATO is load bearing, then either the TLOF or FATO should be marked. Around the FATO and TLOF is the safety area which acts as a buffer. There does not need to be a solid surface for the safety area, however it should remain clear of obstacles that can interfere with the landing of the helicopter. The minimum size of these zones will be determined by the largest rotor diameter of the helicopter intended to land on the pad. Figure 1 shows the overall structure of a helipad.

Figure 1

Structure of a Helipad



Helipads are also a part of some form of landing facility. There are 4 main types of facilities that will contain a helipad. The first type is a heliport, which will have services for helicopters, such as refueling and repairing. The second is a helistop, which is a landing area that offers no services, and allows for clearly marked landing areas

without the need to make a heliport. The third is an Emergency Helicopter Landing Facility and this is effectively a helistop that is only used in the event of an emergency. The last type is a helideck, which is a landing facility over the water, such as a boat or oil rig. Along with these facilities we will also consider parking pads and helicopter runways as a helipad for our purposes as they strongly imply that the area is intended for helicopter to land and takeoff at.

A-2 Related Work in Helipad Identification

There are two main types of approaches that can be taken for identifying helipads. A model-based approach can be developed, which involves having a human determine what features should be present at a helipad and then classifying images based on the presence of these features. The advantage of model-based approaches is that the reason for their process is explainable, and these systems can be made with very little data. Data-driven algorithms involve the collection of large amounts of data, and then implementing an algorithm that will determine the features present at the helipad and will search images for these features. These systems can identify very complex patterns. However, a drawback to these algorithms is that they are very data hungry and require more computational resources.

Most of the known algorithms that have attempted helipad detection have been model-based systems using computer vision approaches. [2] is able to identify a helipad using only the “H” marking. The Speeded-Up Robust Features algorithm is used to propose candidate matching points, and these points are further refined to feature descriptors. These points are then compared to points on an “H” template to determine

the similarity of the template and the ariel image. However most other methods first identify the circle around the “H” when identifying helipads. [3] starts by using a Hough transforms so that circles can be identified. Once a circle has been identified the area needed to be searched can be reduced to areas with the circle. To find the “H” marking the algorithm proposes 12 corners within the circle and if these corners meet 3 criteria, then these corners are likely the 12 corners on an “H”. Another approach explored in [4] starts with creating blobs of connected pixels. Using the Euler number, these blobs can be filtered by blobs that have 0 holes (like an “H”) or 1 hole (like a circle). The 1 opening group is further filtered by the blob’s eccentricity which measures the deviation of a curve from a circular curve. Then the center of the blobs from both groups are compared so that the center of a circle-like blob should be in the same place as the center of 0 opening blob. To help ensure this is a helipad, the ratios of the shapes’ perimeter and area are checked against the expected ratios.

These algorithms have some restrictions though. First, they have only been shown to work in simple environments and may not work in more complex environments. Secondly, these algorithms have limited effectiveness at further distances and angles. These issues were addressed in [5]. This algorithm starts with a system to reduce the complexity of the image by either identifying and removing empty areas (like the ocean) and breaking the image into sub-images. Then to identify an ellipse it finds the edges in the image and, chooses 3 edge points and attempts determine if the points are part of the same ellipse using a property of tangent lines of conics. As the 3 edge points may not be part of the same object, this process is repeated to increase the chances of the ellipse being found. Depending on the distance, the algorithm may then identify the center “H”

based on the presence of 2 parallel lines connected by another line, or if the “H” is not clear the algorithm may use other shape descriptors to determine if the center is approximately an “H”.

While these methods can identify helipads that are adhering to the recommended standard set in the FAA’s 150/5390-2C [1], neither the circle nor the “H” is a requirement even when these standards are adhered to. There are many possible cases, and a lot of scenarios would have to be considered to handle this detection task using a model-driven algorithm. However so long as examples of these helipads that do not adhere to the recommended standards can be found, they can then be used in a data-driven algorithm so the algorithm will learn to identify these features and identify other patterns of helipads that may be difficult for humans to describe. While to our knowledge no peer reviewed paper has been written on data-driven approaches to identify helipads, HelloPad [6] is a system that uses a data-driven algorithm to identify helipads within a specified region. The system uses a sliding window and a trained a ResNet model to identify if a helipad exists at a given location. For the test in Los Angeles the precision and recall of HelloPad reported were 67.2% and 90%, respectively. However, HelloPad collected negative examples from urban settings, and will likely not transfer well to all areas of the U.S.

A-3 Explainable Deep Learning for Computer Vision

A key feature of a deep neural network is its ability to perform representation learning. Representation learning allows for the network to do feature learning, which allows for feature engineering to be done by a neural network. Previously feature

engineering was an important step in processing data for any algorithm handling complex data. For the case of a picture, features from the picture would need to be identified using techniques such as edge-detection, Speeded-Up Robust Features (SURF), or histogram of gradients. These techniques can be used to pull key features that are used to interpret the image using other techniques such as heuristics, traditional ML, or even shallow neural networks. However, feature engineering requires a lot of domain expertise to identify techniques appropriate for the data, and a lot of time is required to identify which techniques work best especially if multiple techniques need to be combined. A deep neural network allows the first layers to learn simple features, and deeper layers will combine these features to create more complex features. A network with enough depth can then learn complex features that are appropriate for the dataset.

Convolutional Neural Networks (CNNs) are designed to process multi-dimensional arrays such as images. There are 3 layers typically used in CNNs: convolutional layers, pooling layers, and fully connected layers. Convolutional layers consist of filters consisting of learnable weights and are followed by an activation function to introduce non-linearities. By learning values for the weights, the convolutional layer is capable of learning appropriate filters to combine information from previous layers. Pooling layers are typically mixed in with convolutional layers. Pooling layers combine nearby features in an image by down-sampling. Down-sampling has the effect of creating invariance to small shifts, while keeping features in the same relative position. A typical down-sampling method may be keeping the max value, however other down-sampling methods may be used. Fully-connected layers are at the end of the network and will come after the convolutional and pooling layers. A fully-connected

layer will define multiple neurons, and each neuron will assign a weight to each input and combine the weighted inputs. This is typically followed by a non-linear activation function to add in non-linearities for each layer [7].

With many of these layers stacked together, the network is capable of learning very complex functions, however the network is unable to explain the reasoning behind the function. These features may be beyond what humans have discovered, they may only be applicable for the exact data, or they could be remarkably similar to other methods humans use. Unfortunately, there is no easy way to explain the complex functions found by deep neural networks, and this gave rise to the field of explain-ability in ai. One of the more commonly used techniques for explainability are saliency maps. Saliency maps determine where high neuron activity occurred, these neuron activities can then be mapped to the image. Neuron activity is determined in part by the neuron's activation signal, however different algorithms define the activity in different ways.

Chapter 2

Dataset: Acquisition, Curation, and Labeling

II-A Dataset Acquisition

We acquired 3 datasets through the FAA, 1 dataset from the IOWA DOT website, and another dataset from the Arcgis website. These datasets provide longitude and latitude of potential helipad landing locations. We used a google API to extract the corresponding images as well as to sample negative helipad locations. We noticed some discrepancy in the FAA datasets and had to manually curate the coordinates to ensure accuracy for our use cases. In the sequel of this chapter, we elaborate on each dataset, our curation approach, and our collection method for negative samples.

A-1 Google Static Maps API

The service for collecting the imagery was the Google statics maps API, which contains the imagery used in google earth. The service is accessed by sending an HTTP request with a query containing the desired parameters, which will be responded to with an image based on the parameters. The parameters used are center, zoom, size, and maptype. Center determines the coordinates that should be the center of the image. Zoom determines the distance a pixel will represent. Size determines the number of pixels in the image. Maptype determines which type of image should be retrieved (as Google maps contains road maps). For the purposes of this project, size was always set to the maximum value of 640x640, and the maptype was always satellite. Center was set to the desired coordinates to be sampled for the image and depending on the case can represent either a helipad location or a location where a helipad is not expected. This will be set to

the coordinates of the area to be sampled. Zoom was later fixed to 18 after a few tests. The most detailed images are at a zoom of 20, however a zoom of 18 was used instead. The difference between the two zooms can be seen below in Figure 2. A zoom of 18 was chosen as the larger area gives a larger margin of error for coordinates and will allow for sampling of larger areas using fewer API calls. There is a cost associated with making API calls beyond a certain limit, so efficiency of calls will become important when scaling up.

Figure 2

Difference in Zoom Levels

(1) : Zoom Level 18 (2) : Zoom Level 20

A-2 Helipad Datasets

Areas with helipads are needed to create the positive set of the dataset. While areas can be randomly sampled and helipads in those areas labeled, this would be

incredibly inefficient. This is because there is an extremely low probability that a randomly sampled coordinate would have a helipad in it. As a result, datasets containing coordinates with helipads are required to collect helipad samples efficiently. Five different datasets were collected, annotated, and used to create a dataset of helipads.

Two of these datasets came directly from FAA databases. The largest dataset came from the FAA's 5010 dataset. This was filtered so that only landing areas appropriate for helicopters were present. However, the 5010 dataset is known to contain errors, which is part of the reason for the necessity of this study. To ensure the accuracy, these coordinates were manually annotated so that only coordinates where a helipad would be visible in the collected image was added to the train set. 6,333 coordinates were in the dataset, however after annotation, only 3,887 were considered to be definitely helipads. Later another large dataset was also provided, however a comparison was done with the first set, and noted a large overlap of locations. The dataset was filtered down to the unique 144 samples found in it. Then the dataset was annotated, and 93 coordinates were considered to be coordinates containing helipads.

Later the FAA had also acquired a dataset from Lifeflight of Maine and provided this dataset as well. This dataset consists of 120 coordinates, which were reduce to 64 usable samples.

Lastly, two datasets were found from online sources. The first source is from a common dataset found online the contains coordinates of hospital helipads around the Los Angeles region. This dataset contained 170 coordinates, and after annotation, 169 of these coordinates were used. Data was also collected from the Iowa DOT's website. The

website listed 126 locations, and 111 of these coordinates were considered to have helipads at them.

A-3 Collection of Non-Helipad (Negative) Examples

Next a negative set images is needed. As databases were needed to find helipads due to the extremely low probability of finding them during random sampling, the negative set was done using random sampling. It is noted that the probability of a helipad existing is non-zero, however a bit of noise is acceptable in the dataset and these helipads can later be identified and removed.

As the goal is currently limited to identifying helipads in the U.S., the sampling was limited to an area such that the sampling region will include most of the mainland U.S. However, most of these samples were of forested areas and farmlands and contained very few urban areas. This could bias the network to expect urban areas to contain helipads. As this is not desired, more urban areas were also sampled so that the negative set would include more urban areas. Different urban areas also have different looks from overhead, so a few different areas were used. It is noted that urban areas will likely have a higher helipad density, and thus a helipad will be more likely to be found there. To lessen this risk, locations like Washington D.C. and New York City were chosen, as after a few helicopter accidents, owning a helipad in New York City has become more difficult, and as Washington D.C. is the capitol, there are increased airspace restrictions making travel by helicopter even more difficult. This causes these cities to have significantly less helipads than expected.

A-4 Issues Encountered During Data Acquisition

On major issue is the reporting error of coordinates. It was decided that for the purposes of this experiment, a coordinate will be considered usable if there was a helipad present in the imagery taken of the area. This is to allow for a margin of error in reported coordinates. The margin is considered acceptable as it is believed to be reasonable for a pilot to identify a helipad within the given area, especially with some prior knowledge. However, there are a few cases of helipads that would be within a reasonable range of the coordinates, however they were not present within the sampled imagery. Figure 3: Non-usable location with helipad nearby, shows a case where there is a helipad near the coordinates, however the helipad is outside the range that was annotated.

Figure 3

Non-Usable Location with Helipad Nearby

While the Google static maps service allows for the easy collection of overhead imagery for most coordinates, there are still a few issues with this service. One of these issues is that this service does not have imagery available at every zoom at every location. Typically, fewer coordinates have available imagery at higher values for zoom.

Another issue with the service is collecting larger imagery. The maximum size of an image is limited to 640x640 pixels. This was the size used during the annotation process, and thus was the size collected. However, while this allows for larger errors in the coordinates, this does also cause an issue when trying to search for and localize

helipad locations. Ideally when searching an image for a 640x640 area that contains a helipad, that image would be larger than 640x640. While a search strategy can be done by probing areas within the search area, due to expected overlap, this is likely to be inefficient in terms of sampling and does not leave a good image to put the results on, as can be done easily on a larger image.

Lastly, there is an issue of recency. The images used in google maps are not real time images, but rather imagery taken during a survey. This means that the overhead view that was sampled does not actually reflect the current state of the area. Google attempts to keep the images up to date such that the available imagery should be less than 3 years old, however that is still a long period of time. This does limit the algorithm's effectiveness for determining the accuracy of new entries of recent helipads, as the available imagery may come from a point before the helipad was constructed.

II-B Dataset and Labeling

B-1 Labelbox Labeling

Labelbox is an online platform that assists in labeling data for image classification, object detection, and image segmentation [8]. Labelbox also supports production pipelines with APIs.

Labelbox was used to manual create bounding boxes around helipads. Bounding box labeling allows us to consider a detection problem beyond a simple binary classification problem (i.e., "Is a helipad present in the image?" – yes or no). It is noted that during the initial labeling, some coordinates had helipads nearby, but no helipad in the image taken. This data was given a more refined look, however instead of labeling

helipad/non-helipad, the labeling was done on Labelbox by placing a box in the area around each present helipad as can be seen in Figure 4. The labeling was also done on larger images by using a collage which uses multiple images to create a single image showing a larger area while retaining the same zoom level. This process is further described in chapter 4. This form of labeling is common for object detection algorithms and will allow for the use of the metrics used by those algorithms to measure the accuracy.

Figure 4

Labeled Helipads

Without the labels in this dataset, the only metric that could be used to define the performance of the network would be the percent of correct predictions. This is a simple classification problem, so every prediction can either be correct or wrong. However, when localizing the object, there are multiple predictions A correct predication also relies on the Intersection over Union (IoU) of the prediction in the label. IoU is determined by dividing the area the boxes overlap by the area that both boxes occupy. If there is no overlap then IoU will be 0, and if the boxes are in the same place then IoU will be 1. If the IoU is above a threshold, then it can be said that the algorithm correctly predicted the existence of a helipad, and correctly determined where the helipad is. This can also be done for multiple helipads in an image.

B-2 Pre-Processing/Data Augmentation

In order to help train the network in a way to prevent memorization, data-augmentation was done. First, in order to alter the dataset to be in the best format for using transfer learning, the images were down sampled to be 256x256 RGB images so that they would be around the input sizes the networks were trained on. ImageNet is a common testbench for architectures, and most architectures have weights trained for ImageNet. Using these weights should help to improve the model's generalization and decrease overall training time. Values for the pixels for the models were also between 0 and 1, and the images collected have values from 0-255. These values were divided by 255 so that they would be in the same range as the image net values. These alterations should make the images more appropriate for possible architectures.

A few other augmentations were also performed to make memorizing the data more difficult. First let us go over the data again. The imagery is taken from satellites that are near the equator. As a result, most images have a bit of shear added because the imagery is not taken from directly overhead. The collected imagery should also always have North on the top of the image, and as we are always in the northern hemisphere, should also cause shadows to be on the South side. The coordinates used are also not exact, so it can be assumed that the helipad may not always be in the middle. As helipads can be facing in any direction, the image may be rotated to any angle and, with the exception of the shadows, should be indistinguishable from normal images. Because the helipads are not always centered, a random translation may also be applied to the image. The translation is limited to be up to 10% of the image to avoid helipads being shifted out of the image. As the imagery will have some shear values, shear was another variable added, and was restricted to be up to 5°. These alterations are intended to prevent the network from memorizing the image during training by memorizing the feature map specific to the image.

II-C Benchmark Dataset

After the above collection, labeling, and curation steps, a helipad identification benchmark dataset is created. The positive set contains 4,324 samples. Some areas are more represented than others, as some of the datasets used were specific to certain regions. However, the largest dataset making up over 80% of the dataset was from the FAA's dataset spread over the United States and its territories and covers different types of landing areas including helicopter parking pads, helidecks, EHLFs, and heliports.

The negative set was created by randomly sampling 5,000 coordinates. 2,000 of these coordinates were from sampling the mainland United States and contains mostly woodland and other rural areas. As urban areas are different and are areas of particular interest, extra sampling was done in these urban areas. 3,000 images were then added to the dataset. These images came from San Jose, Washington D.C., New York City, and San Antonio.

Together these form a dataset of 9,324 satellite images labeled as either helipad image or non-helipad. *Figure 5* shows some of the images in the dataset. (a) shows some of the possible landing locations, including helistops, helidecks, and helicopter runways. (b) shows some of the randomly sampled imagery, with the 3 on the left being samples from the general U.S., and the 2 on the right side coming from the city specific sampling.

Figure 5

Dataset Examples

(a) helipads

(b) non-helipads

Chapter 3

Helipad Classification from Satellite Imagery

III-A Convolutional Neural Networks

The performance of four different networks on this dataset was evaluated. The four networks evaluated were ResNet101 [9], Inception-v3 [10], Xception [11], and EfficientNet-b0 [12]. These networks were chosen to represent as they represent a variety of types of architectures.

A-1 ResNet101

ResNet101 proposed in [9] is a residual network with skip connections. Increasing the depth of a network by adding more can improve the accuracy of a deep neural network, however network performance will begin to suffer if too many layers are added. Skip connections provide an alternative path for learning that can bypass layers which can effectively alter the depth of the network. Doing so allows for the network to be able to act as if it had different depths, as different paths through the network correspond to different depths. This allows for a network to learn an appropriate depth for the dataset it is trained on.

A-2 Inception-v3

Inception-v3, proposed in [10] is part of the family of inception networks. The inception family of networks make use of inception modules. The inception module is made up of parallel convolutional layers that make use of different filter sizes. Larger filters will learn features that cover a wider area, while smaller filters will allow for

learning features that cover smaller areas. Using multiple filter sizes, inception modules are able to learn features corresponding to different filter sizes and combine this information.

A-3 Xception

Xception, proposed in [11], alters the inception-v3 architecture. Skip connections were added to the inception-v3 architecture while the inception modules from inception-v3 were replaced with depthwise separable convolutions. Separable convolutional operations replace the filters using multiple dimensions with filters acting in fewer dimensions. When combined these smaller filters can act as a larger filter while using fewer parameters. Depthwise separable convolutions make use of 1×1×m filters to combine information from multiple channels into a single channel and are combined with information filters that do not combine information from multiple channels.

A-4 EfficientNet-b0

The EfficientNet family of architectures was proposed in [12] and includes eight different architectures (b0-b7). These architectures were designed to efficiently scale, and the higher number architectures correspond to the larger scaled networks. These networks are made up of sections that are organized and repeated with the larger variants containing more of these repeating sections.

Table 1 shows some of the defining features for these architectures and can be used to get an idea of their differences. Note that while the image-net top 5-accuracy does imply that some architectures may be better than others, this is only for the image-net dataset and some architectures may perform better on certain datasets.

Table 1

Summary of Differences Between Models

	Skip connection	Inception Module	Trainable Parameters (millions)	image-net top 5-accuracy
ResNet 101	Yes	No	44.71	92.8%
Inception-v3	No	Yes	23.85	93.7%
Xception	Yes	No	22.91	94.5%
Efficienet-b0	Yes	No	5.33	97.1%

III-B Experimental Results of Convolutional Neural Networks

K-fold validation was used when comparing the performance of these architectures with a value of 10 being used for K. K-fold validation splits the data into K equal subsets. K models will be trained, each using a unique subset for its validation set. Each model will train on all the subsets not used for its validation set. K-fold validation is typically used to ensure that the validation set is not an unfair representation of the dataset, and this is done by creating multiple validation sets and having every datapoint be in the validation set for 1 model. K-fold validation ensures that performance of the network is not based on a validation set made of easier examples, causing the network to appear to perform better than it would on data similar to the training set.

Each model was trained for up to 80 epochs, and early stopping was implemented so that the model will finish training if validation accuracy does not improve after 25 epochs. An epoch is one pass through the entire dataset, meaning that the network can train on each datapoint up to 80 times. The average performance for each architecture is shown below in figures 6, 7, 8, and 9.

Figure 6

ResNet101 Results

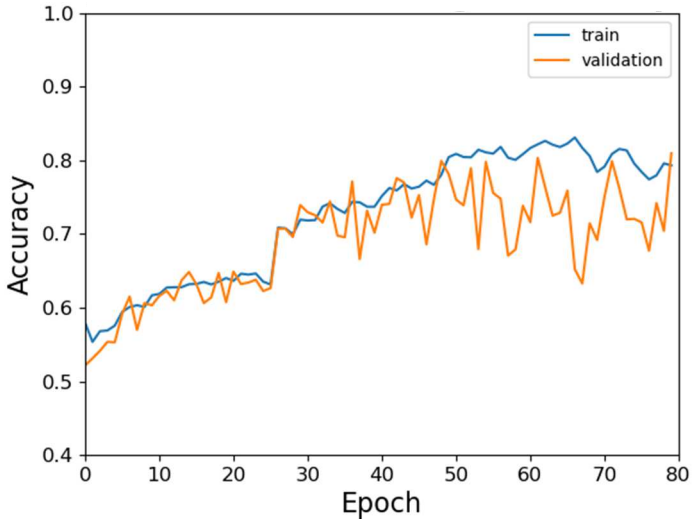


Figure 7

Inception-v3 Results

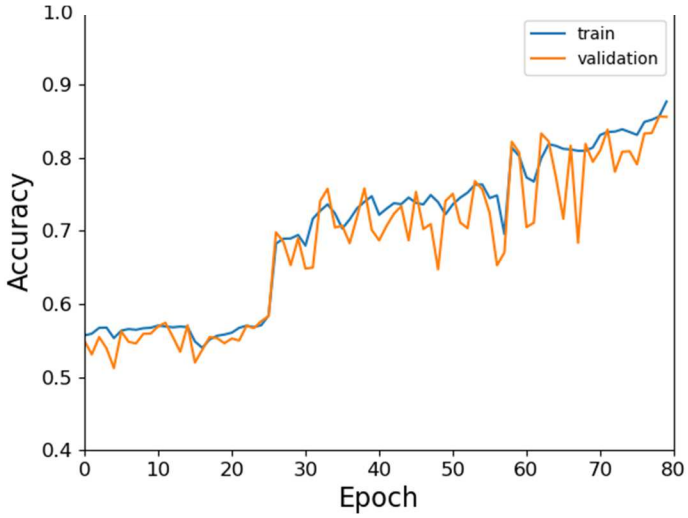


Figure 8

Xception Results

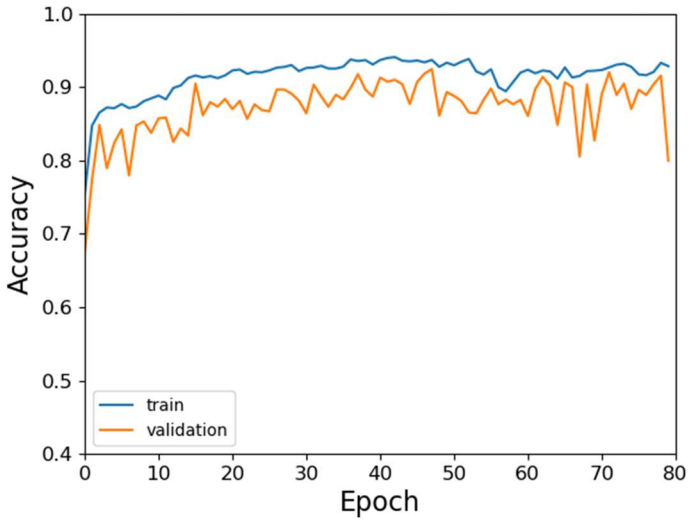
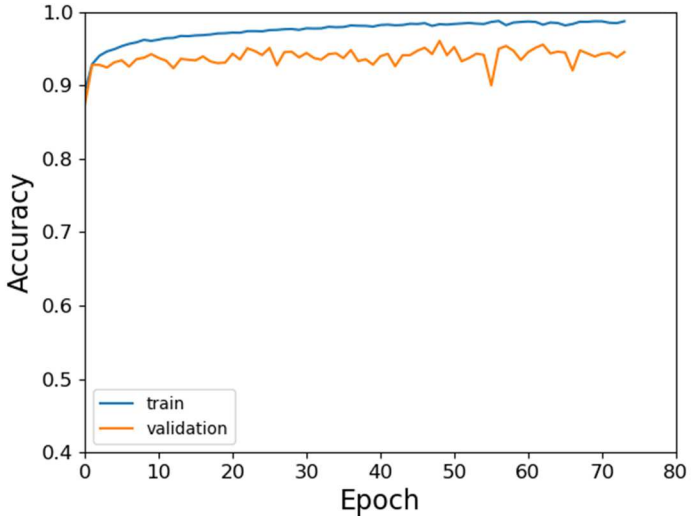


Figure 9

EfficientNet-b0 Results



It can be seen that EficientNet-b0 was able to reach the highest levels of validation accuracy peaking at just above 95% accuracy. However, the shape of the error is also important. There are 2 types of errors, false positives also known as type 1 errors where the network makes a wrongly makes a positive prediction, and false negatives also known as type 2 errors where the network wrongly makes a negative prediction. Accuracy can be a good metric to measure performance if the cost of these errors is equal, however these errors may have drastically different costs associated with them. Below Tables 2, 3, 4, and 5 show the results of these networks as a confusion matrix. Along with the total for each type of error, and overall network performance confusion matrices also show the true positives where the network correctly makes a positive prediction, and true negatives where the network correctly makes a negative prediction. It can be seen the EfficientNet-b0 has both the lowest number of false positives and lowest

number of false negatives. As using EfficientNet-b0 will minimize both the occurrence of false positives and false negatives, it would have the lowest cost regardless of how the costs are weighted.

Table 2

ResNet101 Confusion Matrix

	Prediction : No Helipad	Prediction : Helipad
Label : No Helipad	3768 (45%)	732 (9%)
Label : Helipad	1826 (22%)	2069 (25)%

Table 3

Inception-v3 Confusion Matrix

	Prediction : No Helipad	Prediction : Helipad
Label : No Helipad	3001 (36%)	1499 (18%)
Label : Helipad	1763 (21%)	2132 (25%)

Table 4

Xception Confusion Matrix

	Prediction : No Helipad	Prediction : Helipad
Label : No Helipad	3885 (46%)	615 (7%)
Label : Helipad	269 (3%)	3626 (43%)

Table 5

EfficientNet-b0 Confusion Matrix

	Prediction : No Helipad	Prediction : Helipad
Label : No Helipad	4301 (51%)	199 (2%)
Label : Helipad	212 (3%)	3683 (44%)

III-C Explainability for Convolutional Neural Networks

However, while the performance on the dataset is high, this is not a guarantee that the network is working as intended. The images were taken at a distance where a lot of the area around the helipads were present and helipads are more likely to be found in near certain areas rather than others. As the negative sampling was restricted to certain regions, there is a chance that the network is making its predictions based on features that correlate with the presence of a helipad rather than the actual presence of a helipad. Explainability algorithms would allow for us to determine what areas the network uses when making a prediction and ensure that the network is using the presence of a helipad when making its predictions.

One of the more common methods used in explainability is saliency maps, which show which neurons were important in the network making its prediction. As these neurons also correlate with specific regions of an image, these saliency maps can be stretched to fit over an input image. The algorithm used to generate a saliency map was grad-cam [13]. Grad-cam determines the importance of a neuron by combining the activation information of the neuron with the update information of the associated weights created from training the network on the image. Figures 10, 11, 12 show the grad-cam samples results of an image with a helipad where the network correctly

predicted that there was a helipad, a case referred to as a true positive. In these images the highest neuron activity corresponds to sections with a helipad or is very close to a helipad. This allows for confirmation that the network is predicting that the image contains a helipad based on the presence of a helipad rather the presence of nearby features such as structures or pavement that typically correlate with nearby helipads. The mapping generated will also allow for a human auditor to quickly identify where in the image a helipad is should such a step be necessary. Figure 13 shows the case where there is no helipad, but the network incorrectly guessed that there was a helipad, a case referred to as a false positive. Using grad-cam we can identify what area caused the network to make this prediction. In figure 13, we can see that the activation corresponds to an interesting feature where the roadway widens for a short segment. Noting that the network will fail in such a case, corrective measures can be taken so that the network will be more likely to learn that such areas are not helipads. More grad-cam examples can be seen in Appendix A.

Figure 10

Grad-CAM TP Sample 1

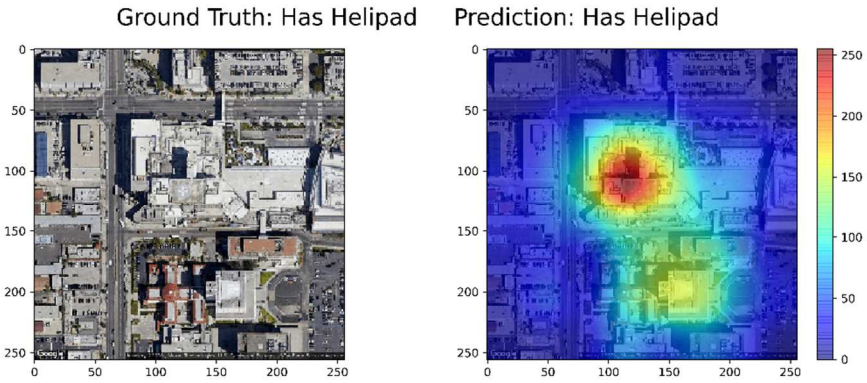


Figure 11

Grad-CAM TP Sample 2

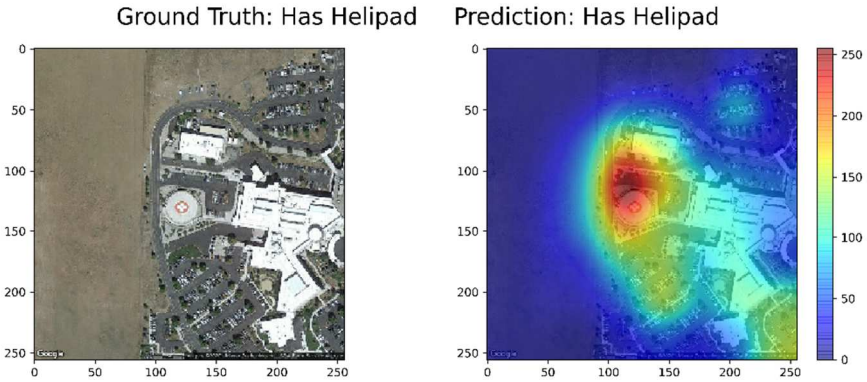


Figure 12

Grad-CAM TP Sample 3

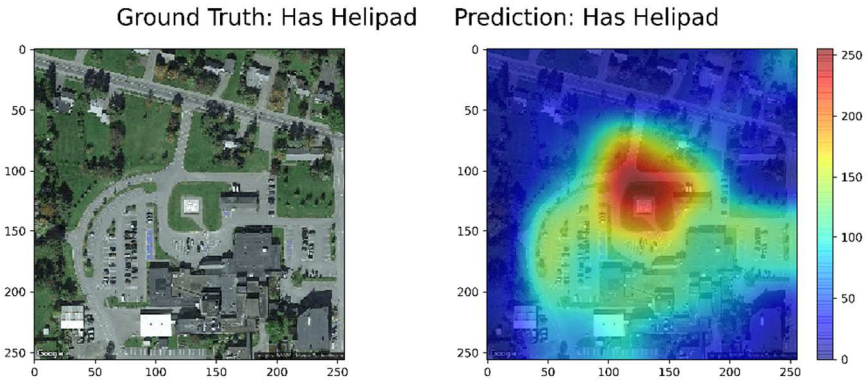
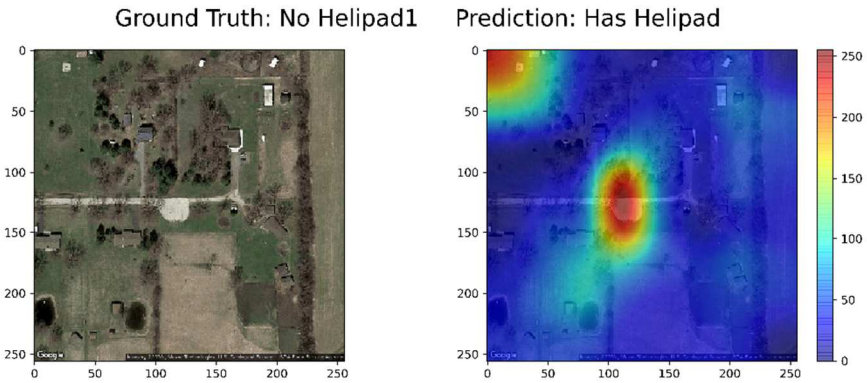


Figure 13

Grad-CAM FP Sample



Chapter 4

Object Localization from Google Earth

The system created thus far can determine if a given set of coordinates contains a helipad. This is useful as this can be used to identify possible errors in a dataset and can be used to check if proposed coordinates contain helipads. However, this algorithm is unable to search for helipads. Adding this feature would allow populated areas to be searched for helipads to add to the dataset. If the system can start searching for helipads, this also means that when a coordinate is flagged as being inaccurate, the system can then search the nearby area and correct coordinates that may be slightly off. Adding this feature will extend this system so that along with removing coordinates from the dataset, the system would be able to suggest new coordinates to add to the dataset.

There are two options for collecting larger imagery for scanning regions using the Google static maps API. The first option would be to use a lower zoom value so that larger areas can be sampled. While this allows for sampling large areas using fewer API calls, there will be less detail for the input images to the network. The second option would be to sample the region multiple times and combine this information. While this method is not very sampling efficient the images will have the same detail as used by the network and would be appropriate to be implemented with the current system.

IV-A Collage

An efficient way to sample the region would be to apply a sliding window over the area, however, a mapping needs to be determined to map the pixels to lat/lon coordinates to do this efficiently. This mapping will allow for sampling the area

corresponding to the adjacent pixels. A mapping was found that maps pixels to meters using the latitude and the zoom level as parameters. This equation is shown in eq 1. This allows for measuring the real-world length of objects by counting the pixels. Common navigation equations can also be used to map meters to change in lat/lon as shown in eq 2 and eq 3. These can be combined with eq 1 to determine the change in lat per pixel and change in lon per pixel as shown in eq 4 and eq 5, respectively. As the desired coordinates are at the center of the image, we can use eq 4 and eq 5 to determine the coordinates of each pixel in the image. Using this the appropriate next coordinate can be determined for a sliding window such that an image corresponding to the desired pixels can be sampled. This allows multiple adjacent images to be sampled and combined into what is referred to as a collage.

$$\frac{meter}{pixel} = 156543.03392 * \frac{\cos(latitude * \frac{\pi}{180})}{2^{zoom}} \quad (1)$$

$$\frac{\Delta latitude}{meter} = \frac{1}{111320} \quad (2)$$

$$\frac{\Delta longitude}{meter} = \frac{1}{111320 * \cos(latitude * \frac{\pi}{180})} \quad (3)$$

$$\frac{\Delta latitude}{pixel} = \frac{\Delta latitude}{meter} * \frac{meter}{pixel} = 156543.03392 * \frac{\cos(latitude * \frac{\pi}{180})}{2^{zoom} * 111320} \quad (4)$$

$$\frac{\Delta longitude}{pixel} = \frac{\Delta longitude}{meter} * \frac{meter}{pixel} = 156543.03392 * \frac{1}{2^{zoom} * 111320} \quad (5)$$

The collage creates a larger image around the desired coordinates using the sliding window. For the implementation, a radius can be set, and then the collage will create an image of $(2\{radius\}+1)^2$ images. To cover as large areas as possible using as few API calls as possible, an image size of 640 x 640 is used for these collages as this is the maximum size. An overlap of 5% is in the sliding window for these collages to both remove the label on the bottom attached by google, and to give room for other algorithms to align the images should it be necessary. These parameters result in collages of size $[(2\{radius+1) \times 608 + 32\}$ x $[(2\{radius\}+1) \times 608 + 32]$ pixels.

IV-B Sliding Window Approach

To scan this area for helipads, another sliding window is applied. This sliding window will allow for the collage to be broken into sections appropriate for the algorithm. The advantage of using two sliding windows is that the area can be efficiently sampled using the first window and searched more thoroughly in the second sliding window. To reduce repetition of pixels during sampling, the first sliding window will search a wide area using an overlap of 5%. The second sliding can search a wide area using a larger overlap such as 50% to increase the repetition of pixels. In the case of 50% overlap in both the x and y directions, each pixel that is not on the edge of the area will be represented in four different images. Another note for this overlap is that each pixel will have one image where they are in the center half of the image in both the x and y directions. This guarantees that even if the helipad is on the edge for three images, it will be in the center for at least one image. Figure 14 shows an example where a helipad is cutoff in some images but becomes centered in the others.

Figure 14

Example of a Partially Cut-off Helipad

Note. The 50% overlap ensures that a helipad will be nearly centered for 1 image, which is the case for the top right image.

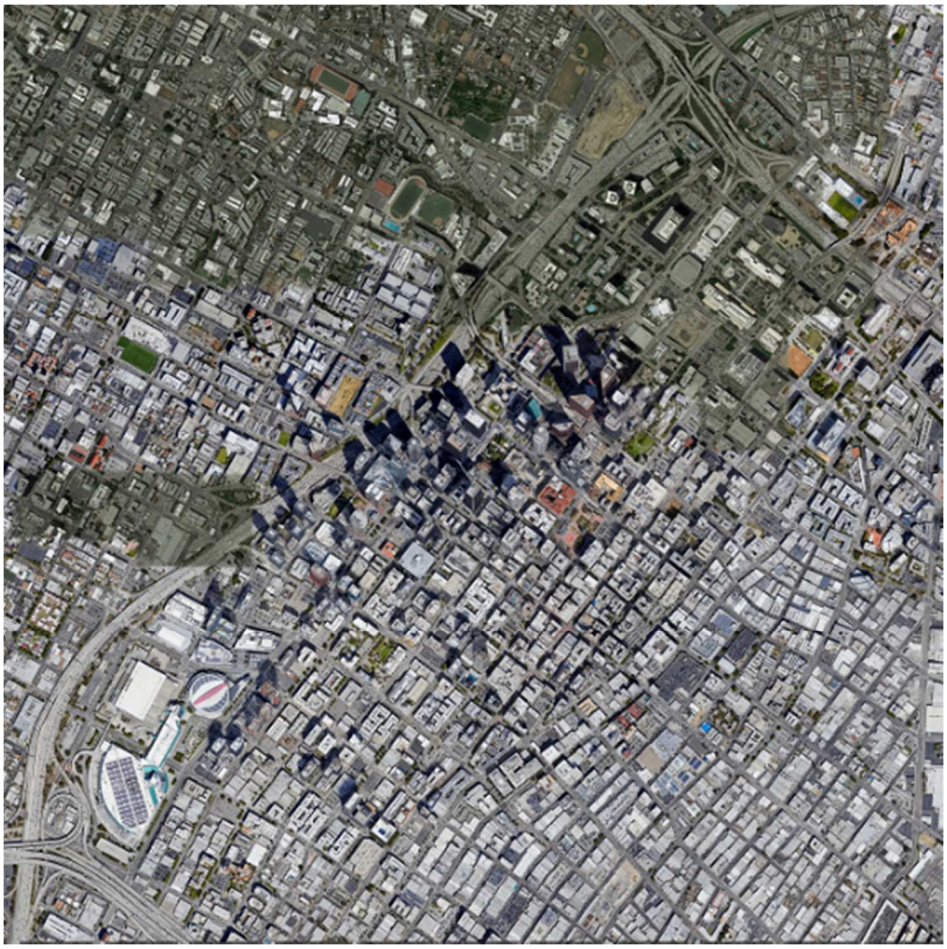
IV-C Experimental Results

We applied this search strategy to a region of LA that is shown in Figure 15. LA was chosen as it is a city with a particularly high helipad density. However due to the

high helipad density this area was not chosen for the negative sampling, and the negative sampling done may not accurately represent this area. To fix this imbalance, the dataset was supplementing by taking the top half of the sampled region and using this half to supplement the training set.

Figure 15

Sampled LA Region



The LA area under study was formed using an 11 x 11 collage and was broken up using a 20 x 20 sliding window to produce 400 smaller images. The 200 images making

up the top half of the collage will be used to supplement the training set. The bottom 200 images were evaluated on and the results as a confusion matrix can be seen in Table 6, and a visual representation of the results can be seen in Figure 16.

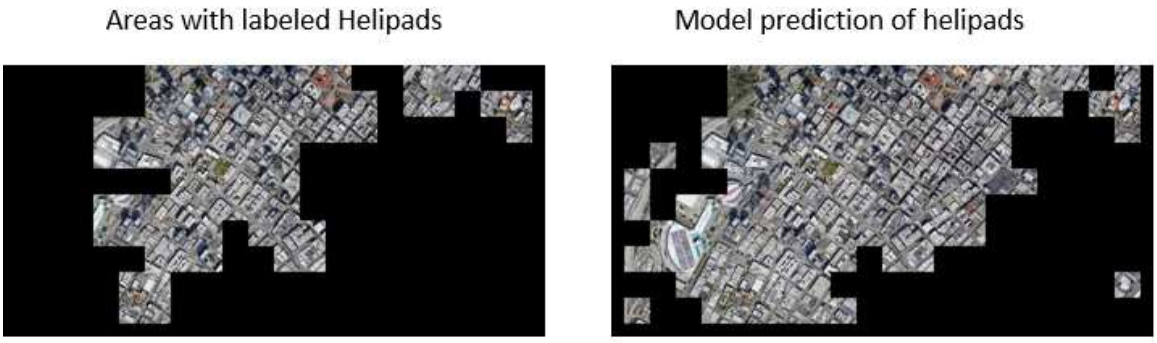
Table 6

Confusion Matrix Results of Area Scan

	Prediction : No Helipad	Prediction : Helipad
Label : No Helipad	78	46
Label : Helipad	2	74

Figure 16

Results of the Helipad Search



Note. Left shows the cells that are labeled as helipads, while the right shows the cells that were predicted as helipads.

Chapter 5

Conclusion and Future Works

V-A Conclusion

While further testing is needed, the proposed system shows capabilities to identify helipads in satellite imagery, determine what area the model used in identification, and search out additional helipads.

In terms of identification, EfficientNet does minimize both the number of false positives and number of false negatives. However, adjustments may still have to be made to further reduce one of these errors, even at the cost of increasing the other errors. A false positive will cause a false confirmation of a helipad, while a false negative will cause the false removal of a helipad. Currently the cost function for the network weighs these errors equally, however the cost associated with these errors are vastly different. Without human confirmation, adding in false positives to the dataset may result in a person attempting to land in an area without a helipad, which can pose a significant risk and causes an unnecessary loss of time and fuel when airborne. However, a false negative will result in helipads not being added to the dataset. Their exclusion will make finding the best helipads more difficult and may result in sub-optimal routing which can result in lost time during emergency scenarios. These costs should be weighted so that the costs are reflective of how much worse one error is compared to another, rather than being treated equally.

The grad-cam implementation allows for the determination of where a helipad is in an image. The use of grad-cam can be helpful for a couple of reasons. The first reason

it that grad-cam does verify that the network is focusing on the helipads in the satellite imagery. The second reason why grad-cam is helpful is the identification of why the network failed. As grad-cam can show where the network was focused when making its predictions, this allows for identifying where the network was looking for the false positive cases. This allows for the identification of cases where the network will fail and allows for corrective measures to be implemented. Grad-cam also highlights the areas of likely helipads, which would help a human auditor to quickly identify where the helipad is that the network found.

Lastly, the system has been extended to be able to search regions for helipads. Despite the CNN being able to achieve high levels of accuracy on its original dataset, it was unable to achieve nearly as high accuracy on the LA region dataset. However, it had very few false negatives meaning that most of the helipads were found. This would allow for the current system to work as a proposal system, and alterations to the dataset may further reduce this number.

V-B Future Work

Some improvements can also be made to this system. The first option would be to increase the areas where negative sampling takes place. It was noted that randomly sampling the U.S. does not provide many urban areas. Four urban areas were added to the sampling pool to increase the representation of urban areas. However, these four areas are likely not representative of many cities in the U.S., This is a possible reason why the algorithm failed in Los Angeles even with supplemental data. Increasing the urban areas sampled may improve the algorithm's ability to generalize to cities.

Grad-cam can also be used to determine the locations of helipads with more precision. The saliency map generated by grad-cam can be used to coarsely locate the object in the image and doing so will allow for bounding boxes to be placed around the proposed region. The center of these bounding boxes can then be calculated, and the coordinates at the center can then be found. This can then also be applied to the region scanning approach to propose helipad locations. The first possible approach would be to propose a helipad location in an image and confirm that the helipad was also proposed in images overlapping the area. The second possible approach would be to combine the saliency maps. This is similar to the first approach; however, this will also make use of all importance values even if they are not near an area which was boxed.

References

[1] 150/5390-2c - heliport design, FAA, 2012.

[2] R. O. Praksh and C. Saravanan, "Autonomous robust helipad detection algorithm using computer vision," in *2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT)*, 2016.

[3] A. Rungta, Y. Soni, P. Agarwal, B. Ghosh and S. Kumar, *Real-time and Autonomous Detection of Helipad for Landing Quad-Rotors by Visual Servoing,* 2020.

[4] C. Patruno, M. Nitti, E. Stella and T. D'Orazio, "Helipad detection for accurate UAV pose estimation by means of a visula sensor," *Internation Jourrnal of Advanced Robotic Systems.,* 09 2018.

[5] Z. Pierre, S. Mavromatis, J. Sequeira, G. Anoufa, N. Belange and F.-X. Fillias, "Embedding intelligent image processing algorithms: the new safety enhancer for helicopters missions," in *44th European Rotorcraft Forum - ERF 2018, S*, DELFT, 2018.

[6] G. Walker, *Detecting Helipads Using CNN,* 2019.

[7] Y. LeCun and Y. H. G. Bengio, "Deep Learning," Nature, vol. 521, no. 7553, pp. 436-444, 2015.

[8] "Labelbox," Labelbox, 2021. [Online]. Available: https:/labelbox.com.

[9] K. He, X. Zhang, S. Ren and J. Sun, Deep Residual Learning for Image Recognition, 2015.

[10] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens and Z. Wojna, Rethinking the Inception Architecture for Computer Vision, 2015.

[11] F. Chollet, Xception: Deep Learning with Depthwise Separable Convolutions, 2017.

[12] M. Tan and Q. V. Le, EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks, 2020.

[13] R. Selvaraju, D. Abhishek, R. Vendantam, M. Cogswell, D. Parikh and D. Batra, "Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization," CoRR, vol. abs/1610.02391, 2016.

Appendix

Sample Grad-CAM Results

Figure A1

Grad-CAM TP sample 4

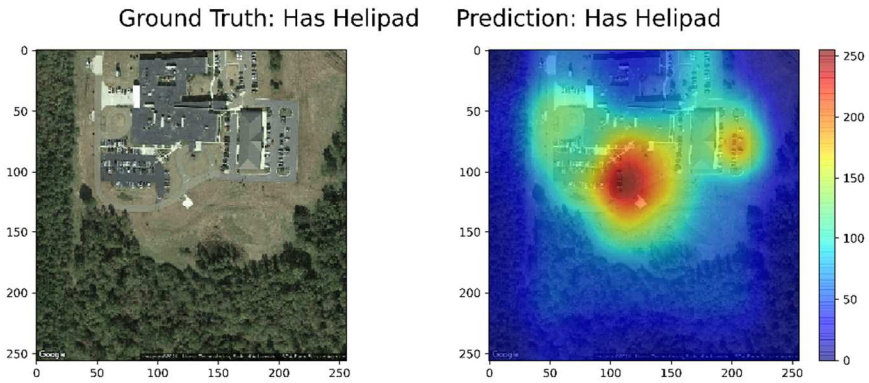


Figure A2

Grad-CAM TP Sample 5

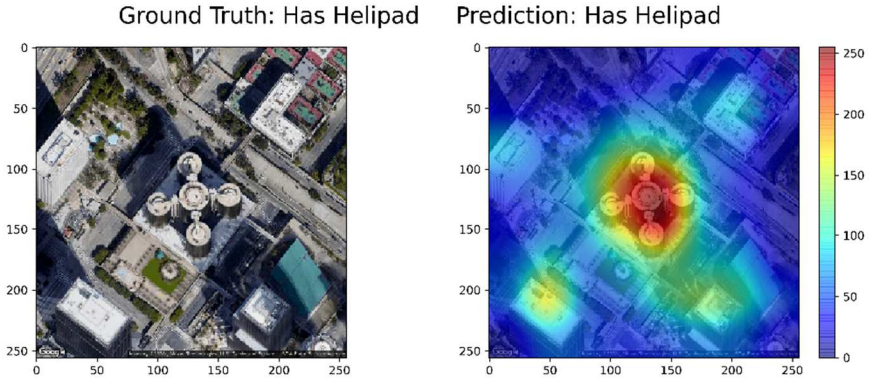


Figure A3

Grad-CAM TP 6

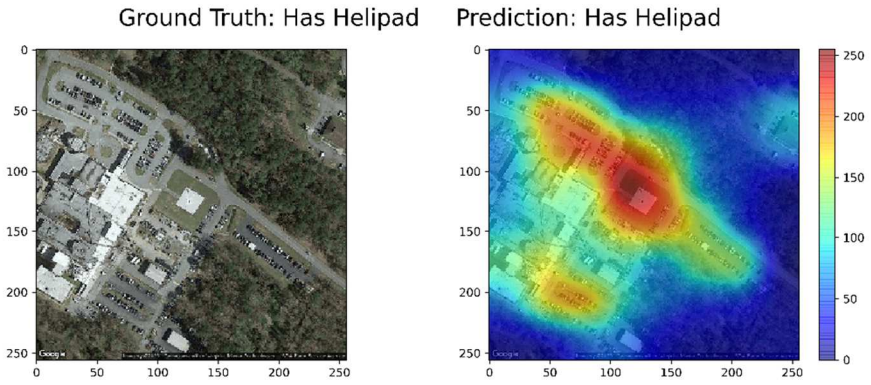


Figure A4

Grad-CAM TP 7

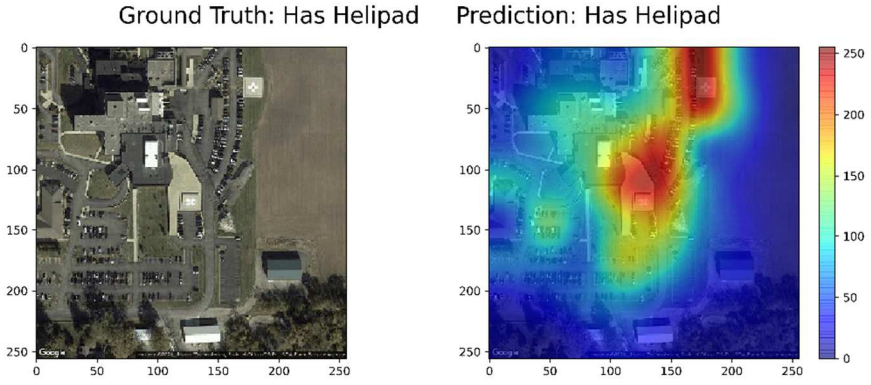


Figure A5

Grad-CAM TP 8

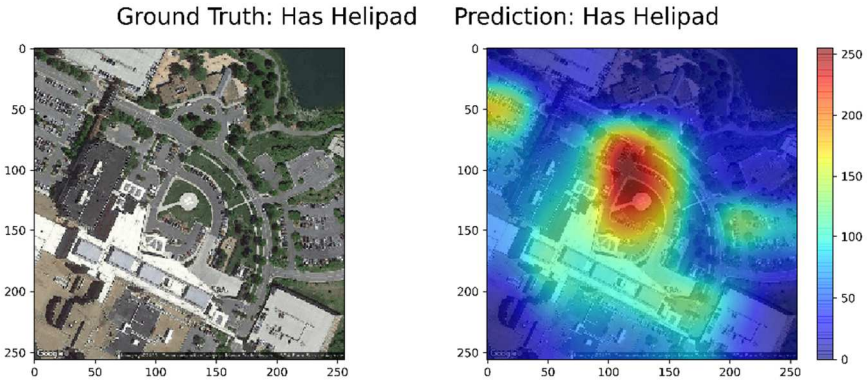


Figure A6

Grad-CAM TP 9

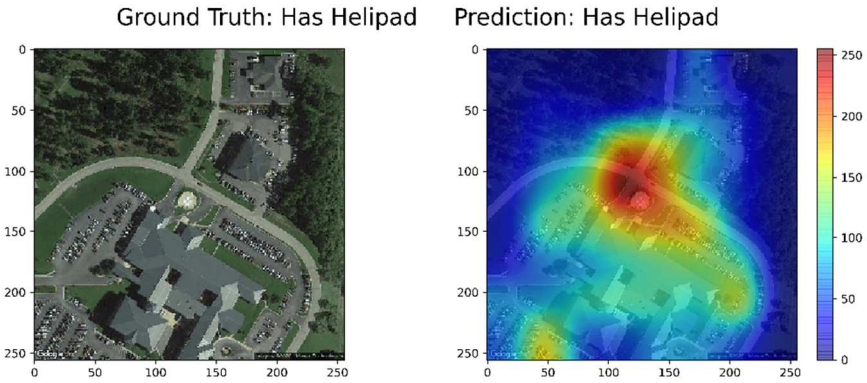


Figure A7

Grad-CAM FP 2

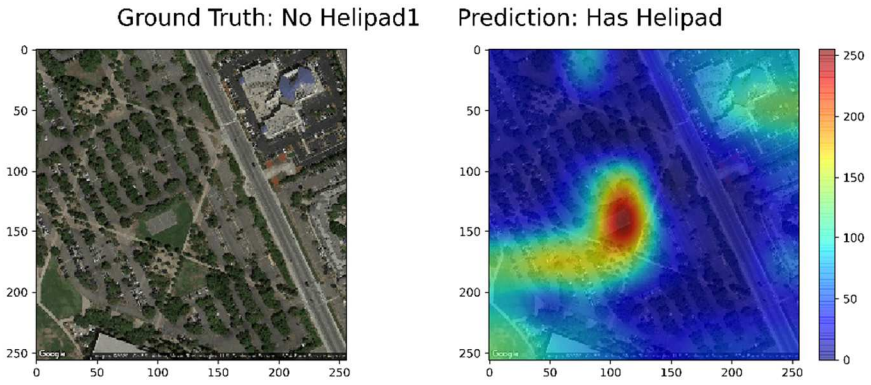


Figure A8

Grad-CAM FP 3